Datum und Zeit

- Datum

Datum

```
//Datum
//Es wird mit Instanzen der Klasse Date gearbeitet
//Erzeugen von Date-Instanzen
Date datum1 = new Date();
Date datum2 = new Date(2015,02,13); //Deprecated
```

Formatieren des Datums

```
//Formatieren des Datums
SimpleDateFormat sdf1 = new SimpleDateFormat("dd.MM.yyyy");
SimpleDateFormat sdf2 = new SimpleDateFormat("yyyy.MM.dd");
String formatiertesDatum = sdf1.format(datum1);
System.out.println(formatiertesDatum);
formatiertesDatum = sdf2.format(datum1);
System.out.println(formatiertesDatum);
```

```
15.10.2015
2015.10.15
```


Parsen eines Datums aus einem String

```
//Parsen eines Datums aus einem String
SimpleDateFormat sdf3 = new SimpleDateFormat("dd.MM.yyyy");
String datumString = "31.12.2015";
Date datum3 = new Date();
try {
    datum3=sdf3.parse(datumString);
} catch (ParseException ex) {
    Logger.getLogger(StringsUndDates.class.getName()).log(Level.SEVERE, null, ex);
}
System.out.println(datum3.toString());

System.out.println(" ");
System.out.println("\n" + "Parsen eines Datums aus einem String Ende " + "\n");
```

```
Thu Dec 31 00:00:00 CET 2015
```

Formatierungen mit DateFormat

```
//Weitere Formatierungen mit DateFormat
Date datum4 = new Date();

//Jetzt wird eine Instanz von DateFormat erzeugt
//Ausgaben mit DateInstance
DateFormat df = DateFormat.getDateInstance();
System.out.println(df.format(datum4));
df = DateFormat.getDateInstance(DateFormat.SHORT);
System.out.println(df.format(datum4));
df = DateFormat.getDateInstance(DateFormat.MEDIUM);
System.out.println(df.format(datum4));
df = DateFormat.getDateInstance(DateFormat.LONG);
System.out.println(df.format(datum4));
```

```
15.10.2015
15.10.15
15.10.2015
15. Oktober 2015
```


Ausgaben mit TimeInstance

```
 //Ausgaben mit TimeInstance
df = DateFormat.getTimeInstance();
System.out.println(df.format(datum4));
 df = DateFormat.getTimeInstance(DateFormat.SHORT);
System.out.println(df.format(datum4));
df = DateFormat.getTimeInstance(DateFormat.MEDIUM);
System.out.println(df.format(datum4));
df = DateFormat.getTimeInstance(DateFormat.LONG);
System.out.println(df.format(datum4));
```

```
05:55:26
05:55
05:55:26
05:55:26 MESZ
```

Angaben mit DateTimeInstance

```
//Angaben mit DateTimeInstance
df = DateFormat.getDateTimeInstance();
System.out.println(df.format(datum4));
df = DateFormat.getDateTimeInstance(DateFormat.SHORT, DateFormat.SHORT);
System.out.println(df.format(datum4));
df = DateFormat.getDateTimeInstance(DateFormat.MEDIUM, DateFormat.MEDIUM);
System.out.println(df.format(datum4));
df = DateFormat.getDateTimeInstance(DateFormat.LONG, DateFormat.LONG);
System.out.println(df.format(datum4));
```

```
15.10.2015 05:59:19
15.10.15 05:59
15.10.2015 05:59:19
15. Oktober 2015 05:59:19 MESZ
```


Calendar & GregorianCalendar

```
//Kalender: java.util.Calendar
//In Europa: java.util.GregorianCalendar
// Default Konstruktor
GregorianCalendar kalender1 = new GregorianCalendar();
//Erweiterter Konstruktor mit Datum und Zeitpunkt
GregorianCalendar kalender2 = new GregorianCalendar(2015,1,1,1,1,1);
```

Setzen von Attributen

```
//Setzen von Attributen
kalender1.set(Calendar.YEAR, 2020);
kalender1.set(Calendar.MONTH, Calendar.FEBRUARY);
kalender1.set(Calendar.DAY_OF_MONTH, 24);

kalender2.set(Calendar.DAY_OF_WEEK, Calendar.MONDAY);
kalender2.set(Calendar.MINUTE, 18);
```


Getter von Attributen

```
//Getter
int jahr = kalender1.get(Calendar.YEAR);
System.out.println(jahr);
int wochentag = kalender2.get(Calendar.DAY_OF_WEEK);
System.out.println(wochentag);
```

```
2020
1
```

Weiterschaltung der Zeit

```
//Methode roll() - Weiterschaltung deer Zeit
kalender1.roll(Calendar.YEAR, 980);
System.out.println(kalender1.get(Calendar.YEAR));
```

```
3000
```


Aktuelle Zeit

```
//Aktuelle Zeit mit Hilfe der Systemzeit und SimpleDateFormat
long timeNow = System.currentTimeMillis();
Date datum5 = new Date(timeNow);
SimpleDateFormat sdf4 = new SimpleDateFormat();
String formatedTimeNow = sdf4.format(datum5);
System.out.println(formatedTimeNow);
```

```
15.10.15 06:22
```

Sprachen und Regionen

- **Locale-Objekte** repräsentieren geografische, politische oder kulturelle Regionen.
- Sprache und die Region müssen getrennt werden
  Region oder Land gibt die Sprache nicht eindeutig vor:

  Beispiel:
  Kanada in der Umgebung von Quebec
  französische Ausgabe ist relevant,
  die unterscheidet sich von der englischen.

Sprachen und Regionen

- Sprach-Objekte werden **immer** mit dem Namen der **Sprache** und *optional* mit dem *Namen des Landes* beziehungsweise einer Region erzeugt.
- Im Konstruktor werden Länderabkürzungen angegeben

```
import java.util.Locale;

Locale greatBritain = new Locale( "en", "GB" );
Locale french       = new Locale( "fr" );

System.out.println(greatBritain);   //en_GB
System.out.println(french);         //fr
```

Sprachen und Regionen

- **Sprachen** werden durch Zwei-Buchstaben-Kürzel aus dem ISO-639-Code[177] (ISO Language Code) identifiziert

  http://www.loc.gov/standards/iso639-2/php/code_list.php

- **Ländernamen** werden Zwei-Buchstaben-Kürzel aus dem ISO 3166[178] (ISO Country Code) identifiziert

  http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/index.html

Konstruktoren der Klasse Locale

- **Locale (String language)**

  Erzeugt ein neues Locale-Objekt für die Sprache (language), die nach dem ISO-693-Standard gegeben ist.

- **Locale(String language, String country)**

  Erzeugt ein Locale-Objekt für eine Sprache (language) nach ISO 693 und ein Land (country) nach dem ISO-3166-Standard

```
//Aktuell eingestellte Sprache
System.out.println(Locale.getDefault());     //de_DE
```

Konstanten für Länder und Sprachen

- Die Locale-Klasse besitzt Konstanten für häufig auftretende Länder und Sprachen
- Konstante für Großbritannien Locale.UK
  statt der Instantiierung new Locale("en", "GB")
- Konstante für Deutschland

```
Locale german = new Locale("de", "De");
System.out.println(Locale.GERMAN);  //de
System.out.println(Locale.GERMANY); //de_DE

System.out.println(german); //de_DE
```


Locale und SimpleDateFormat

```
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
```

```
Date date = new Date();
Locale locHU = new Locale("hu","HU");

SimpleDateFormat locSDF =
        new SimpleDateFormat("EEEE, d. MMMM yyyy",locHU);
System.out.println(locSDF.format(date));
```

```
szerda, 7. december 2016
```

Locale und DateFormat

```
import java.text.DateFormat;
import java.util.Date;
import java.util.Locale;
```

```
Date datum = new Date();
Locale locHU = new Locale("hu","HU");
DateFormat df =
DateFormat.getDateTimeInstance(DateFormat.FULL,DateFormat.FULL,locHU);

String formatiertesDatum = df.format(datum);
System.out.println(formatiertesDatum);
```

```
2016. december 7. 17:46:16 CET
```


Locale und Calendar

```
Calendar calJapan =
Calendar.getInstance(
TimeZone.getTimeZone("Asia/Tokyo"),Locale.JAPANESE);
System.out.println(calJapan.get(Calendar.HOUR_OF_DAY));

Calendar calUS =
Calendar.getInstance(
TimeZone.getTimeZone("America/New_York"),Locale.US);
System.out.println(calUS.get(Calendar.HOUR_OF_DAY));

Calendar calGer =
Calendar.getInstance(
TimeZone.getDefault(),Locale.getDefault());
System.out.println(calGer.get(Calendar.HOUR_OF_DAY));
```

```
2
12
18
```